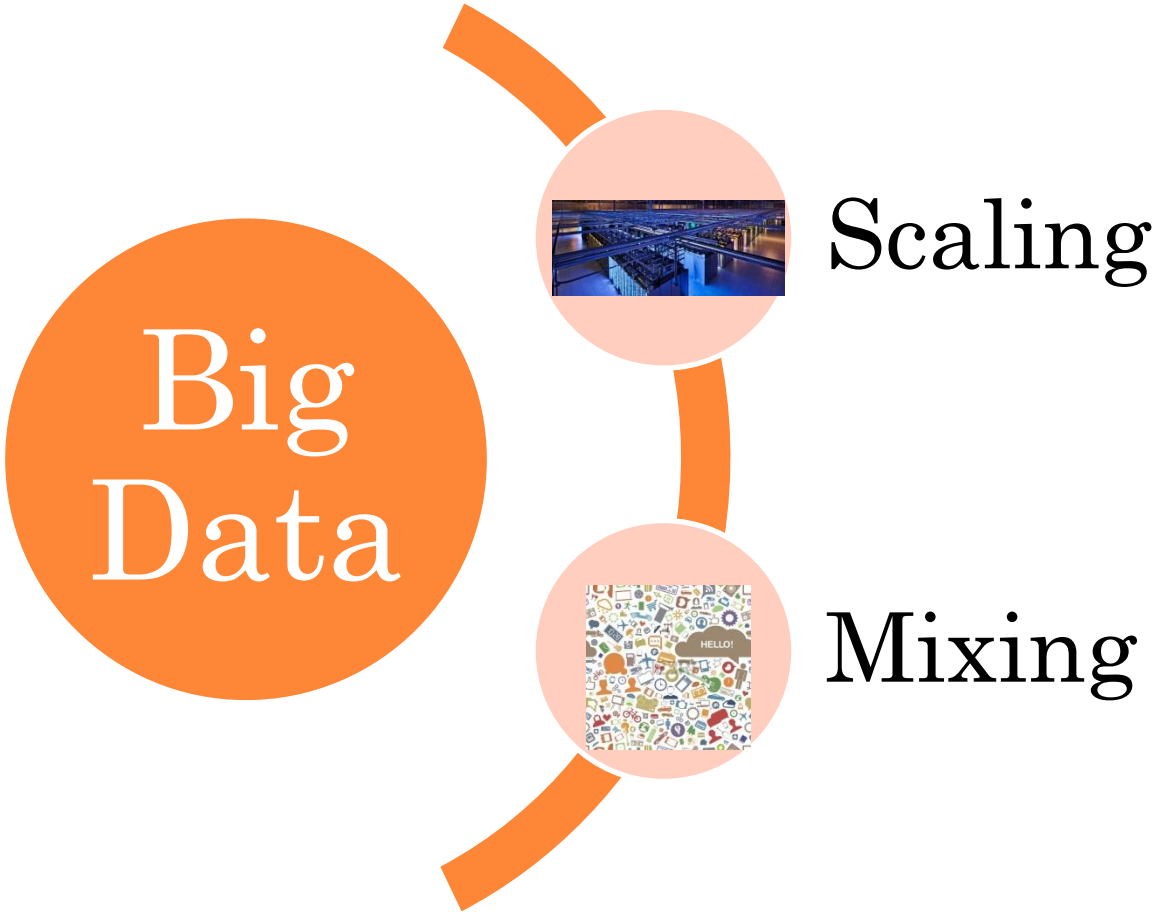
Security and Privacy of Big Data: A NIST Working Group Perspective

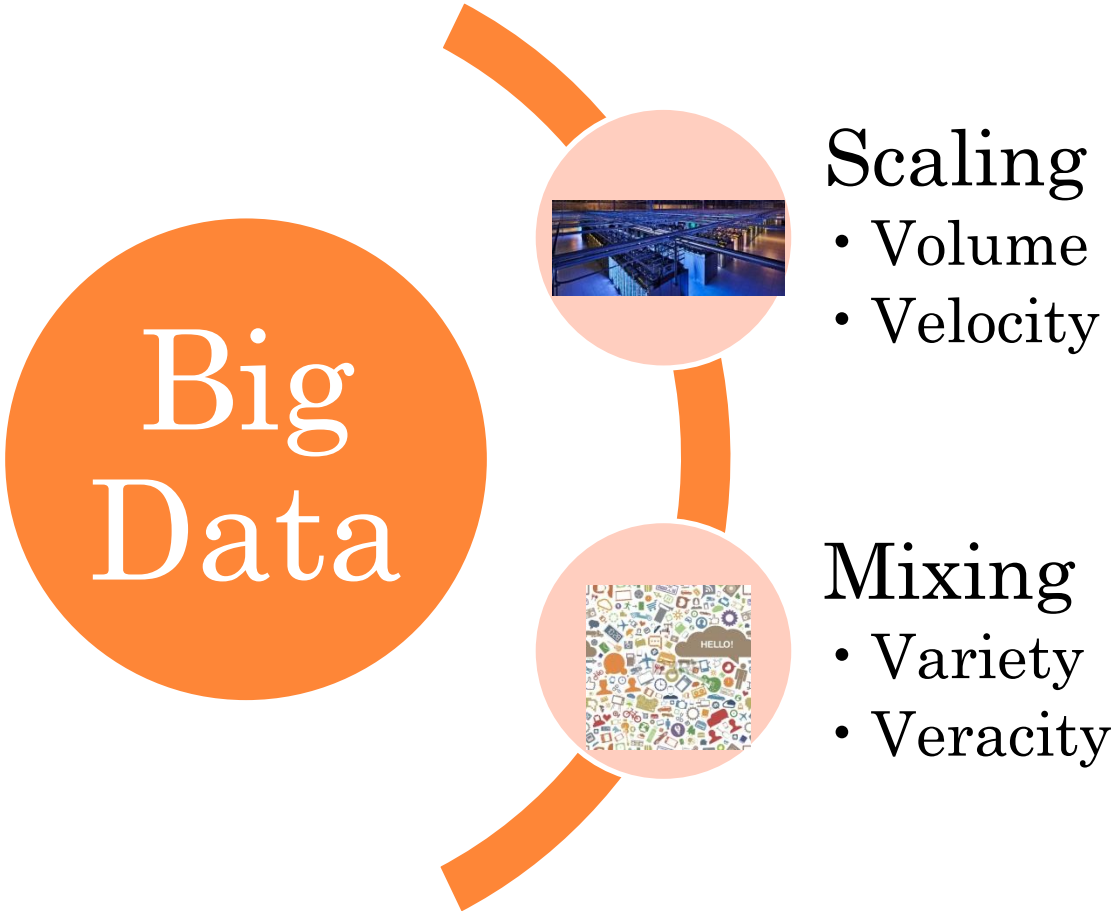
Arnab Roy
Fujitsu Laboratories of America

Co-Chair, Security and Privacy Subgroup
NIST Big Data Working Group

A 10000-feet view



A 10000-feet view



Emergent S&P Considerations

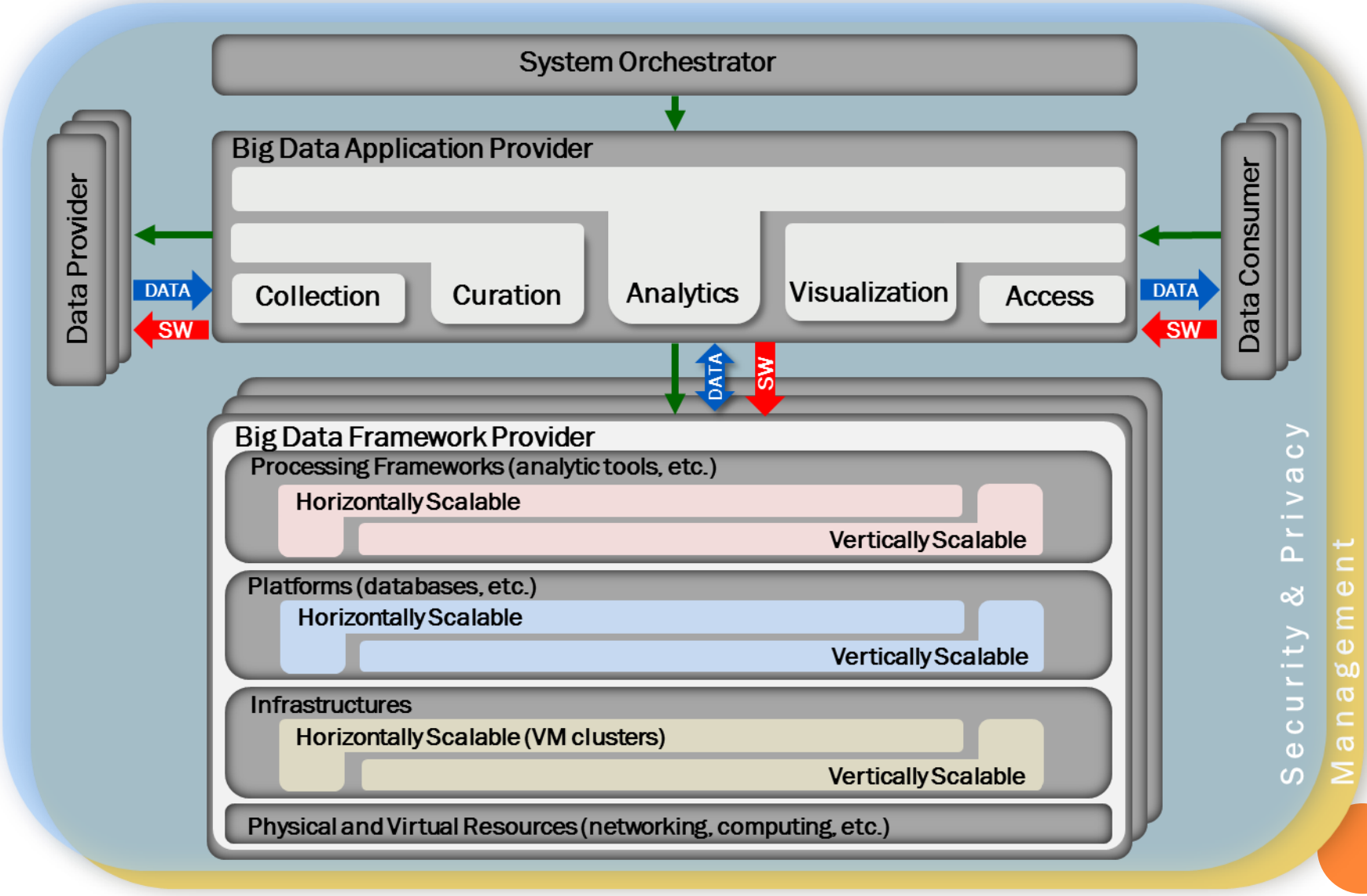
(Big) Scaling - Retarget to Big Data infrastructural shift

- Distributed computing platforms like Hadoop
- Non-relational data stores

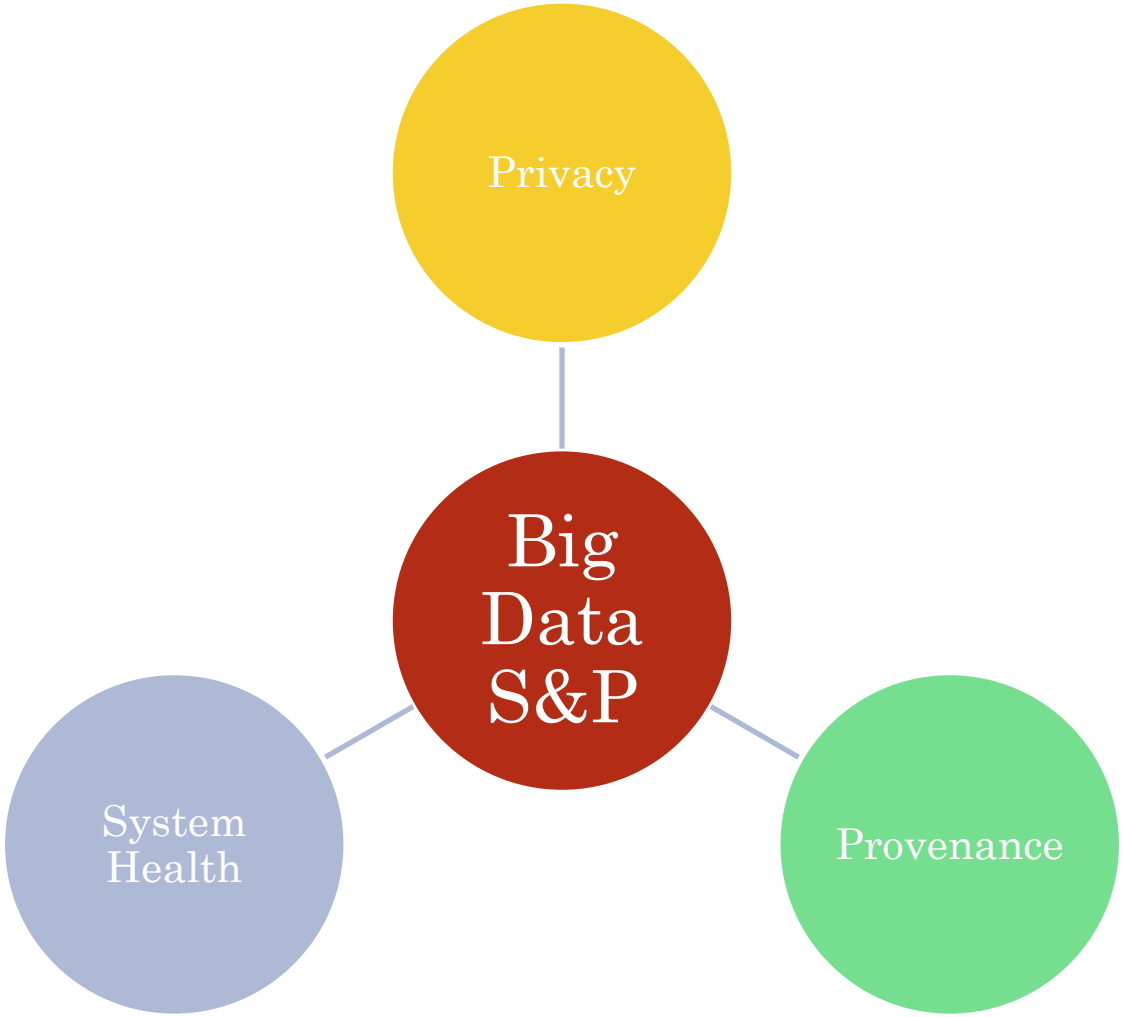
(Data) Mixing – Control visibility while enabling utility

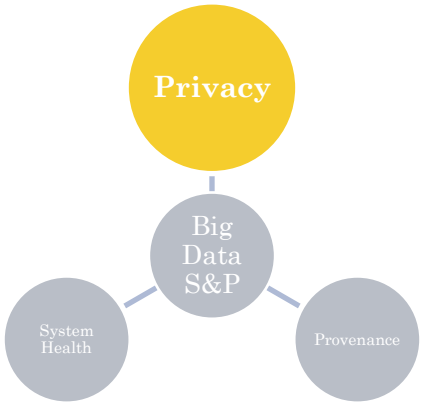
- Balancing privacy and utility
- Enabling analytics and governance on encrypted data
- Reconciling authentication and anonymity

INFORMATION VALUE CHAIN



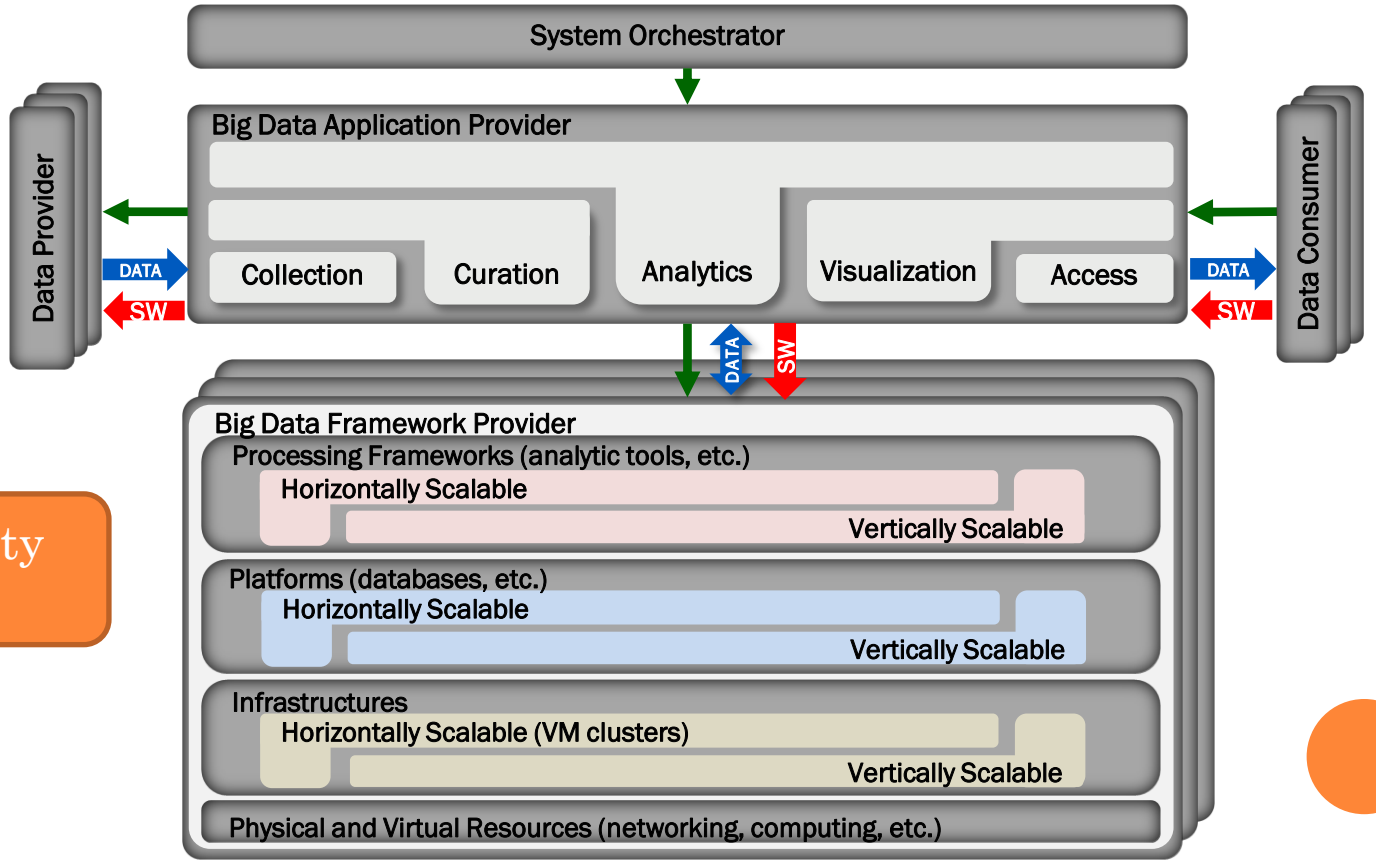
Conceptual Taxonomy of S&P Topics



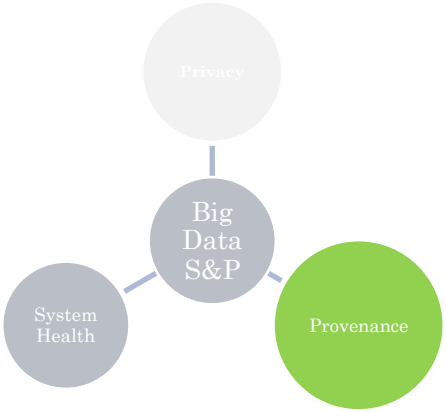


Communication Privacy

Privacy Preserving Dissemination



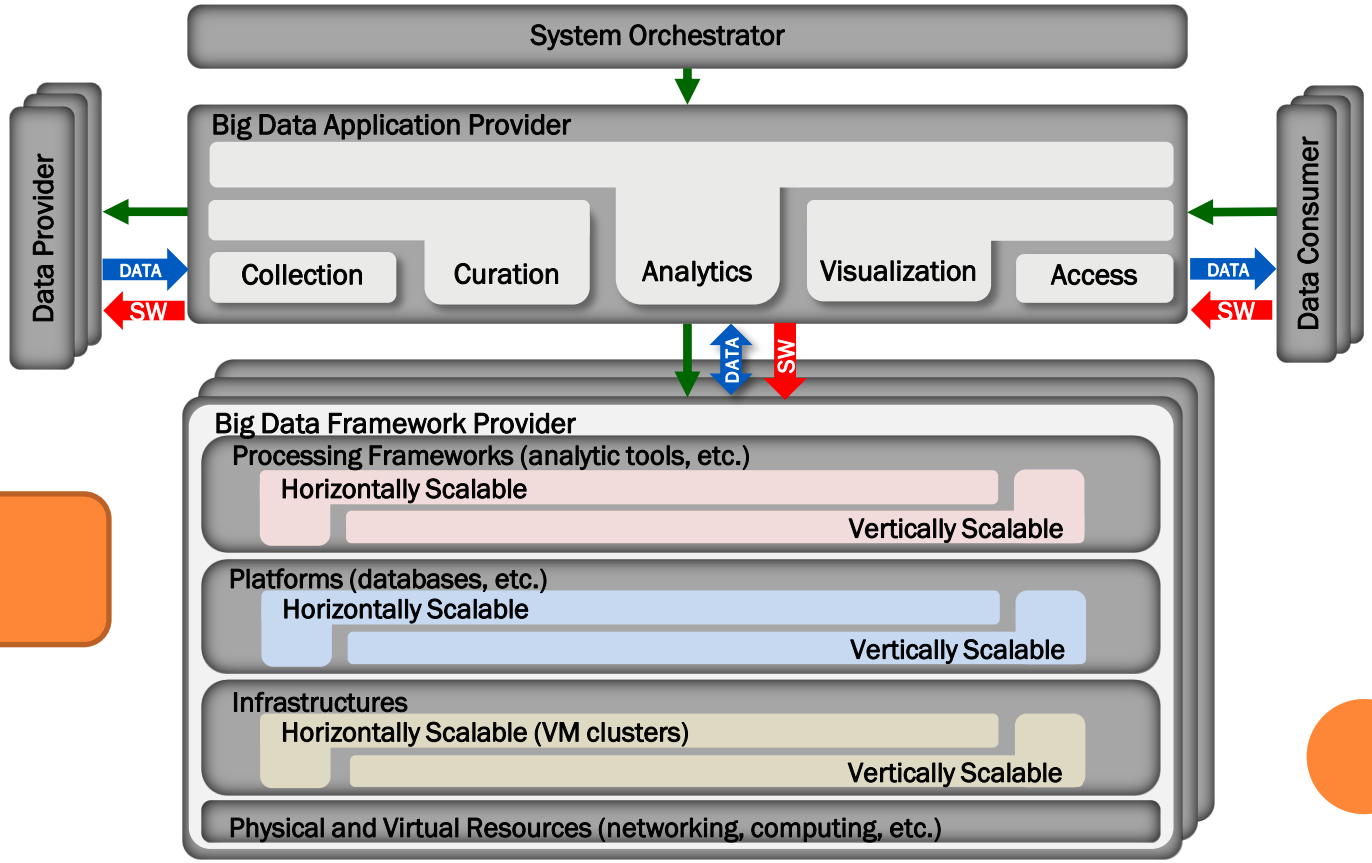
Data Visibility Control



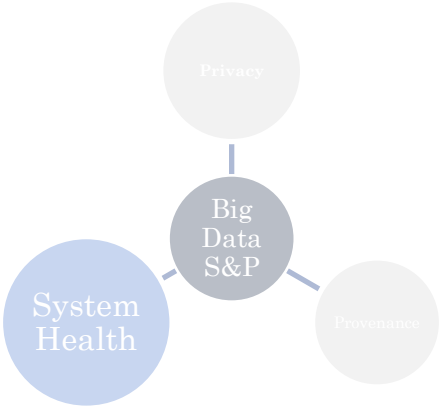
Communication Integrity

Control of Valuable Assets

End-Point Input Validation



Metadata Tracking



Big Data Analytics for Security Intelligence

Resistance to DoS

Resistance to DoS

Forensics

System Orchestrator

Big Data Application Provider

Collection

Curation

Analytics

Visualization

Access

Data Provider

Data Consumer

DATA

SW

DATA

SW

DATA

SW

Big Data Framework Provider

Processing Frameworks (analytic tools, etc.)

Horizontally Scalable

Vertically Scalable

Platforms (databases, etc.)

Horizontally Scalable

Vertically Scalable

Infrastructures

Horizontally Scalable (VM clusters)

Vertically Scalable

Physical and Virtual Resources (networking, computing, etc.)

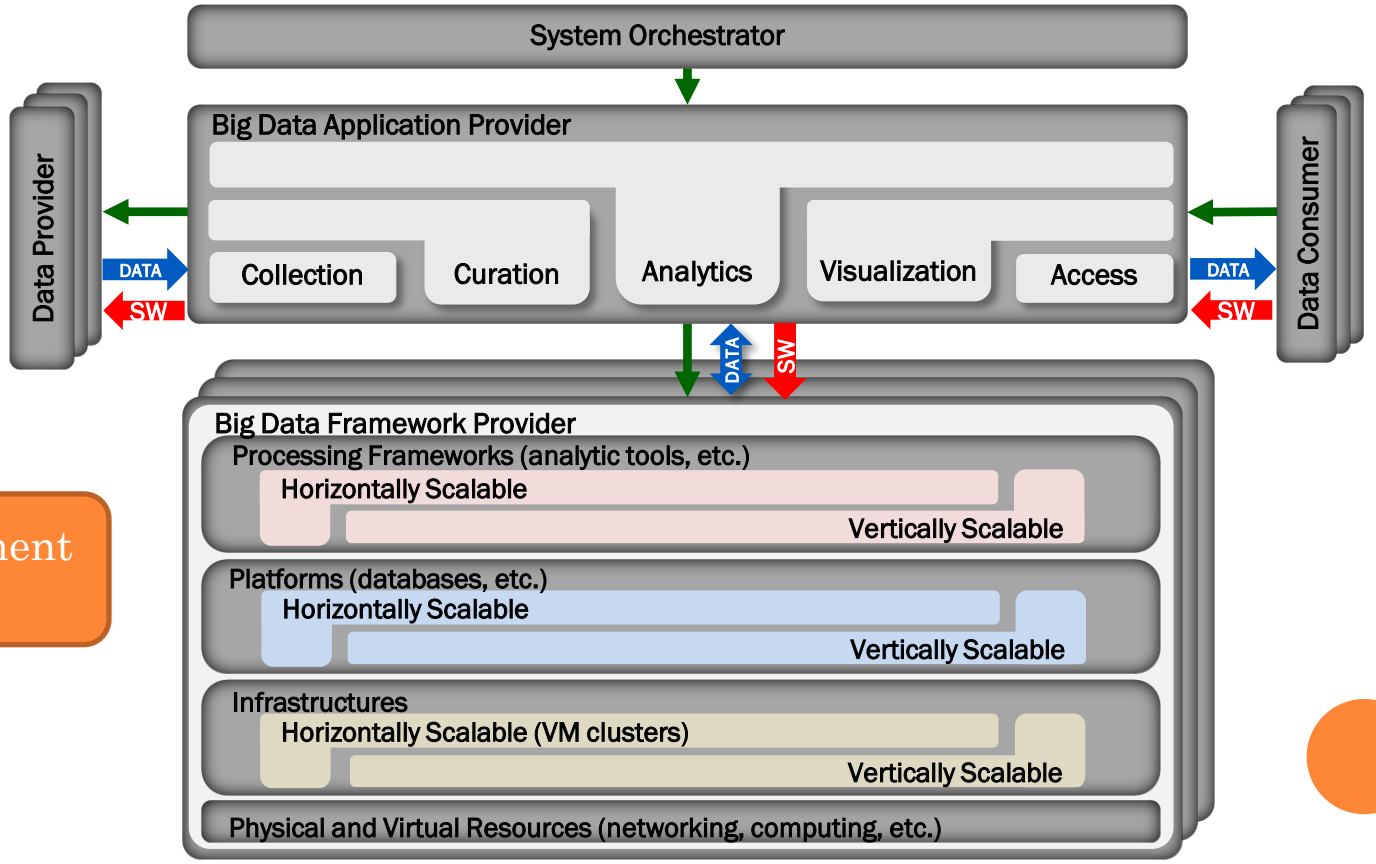
Operational Taxonomy of S&P Topics

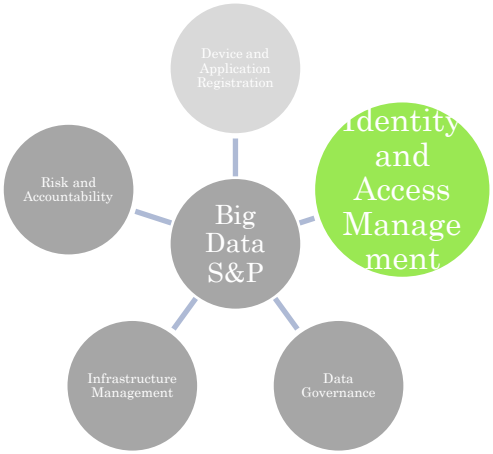
Security Metadata Model

Registration of Applications

Registration of Devices, Users

Policy Enforcement and Audit

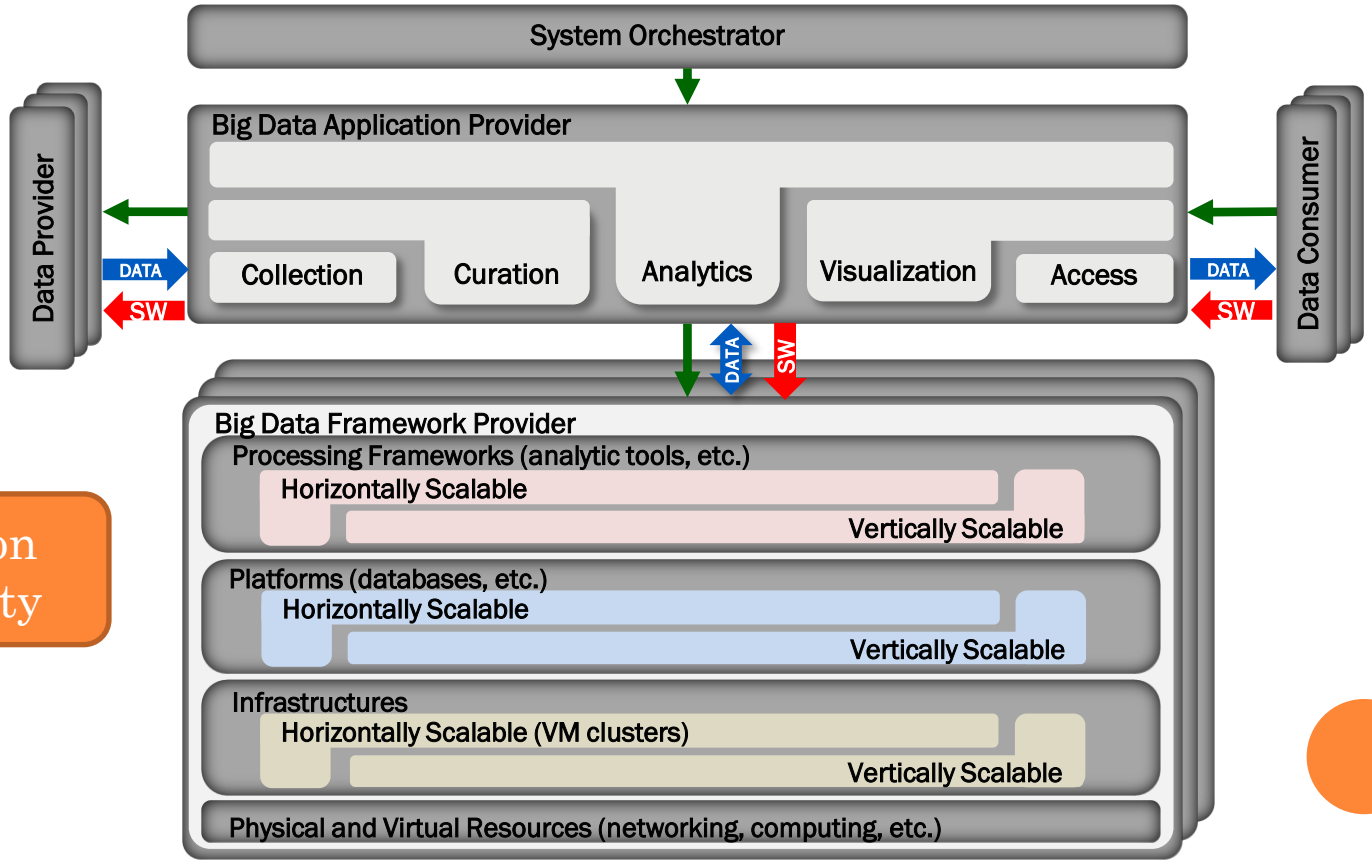


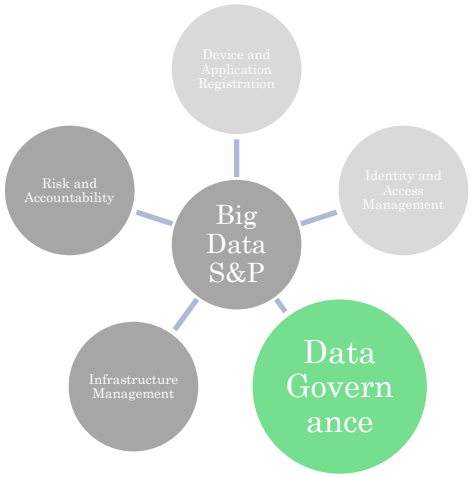


Application Layer Identity

Identity Providers

Virtualization Layer Identity

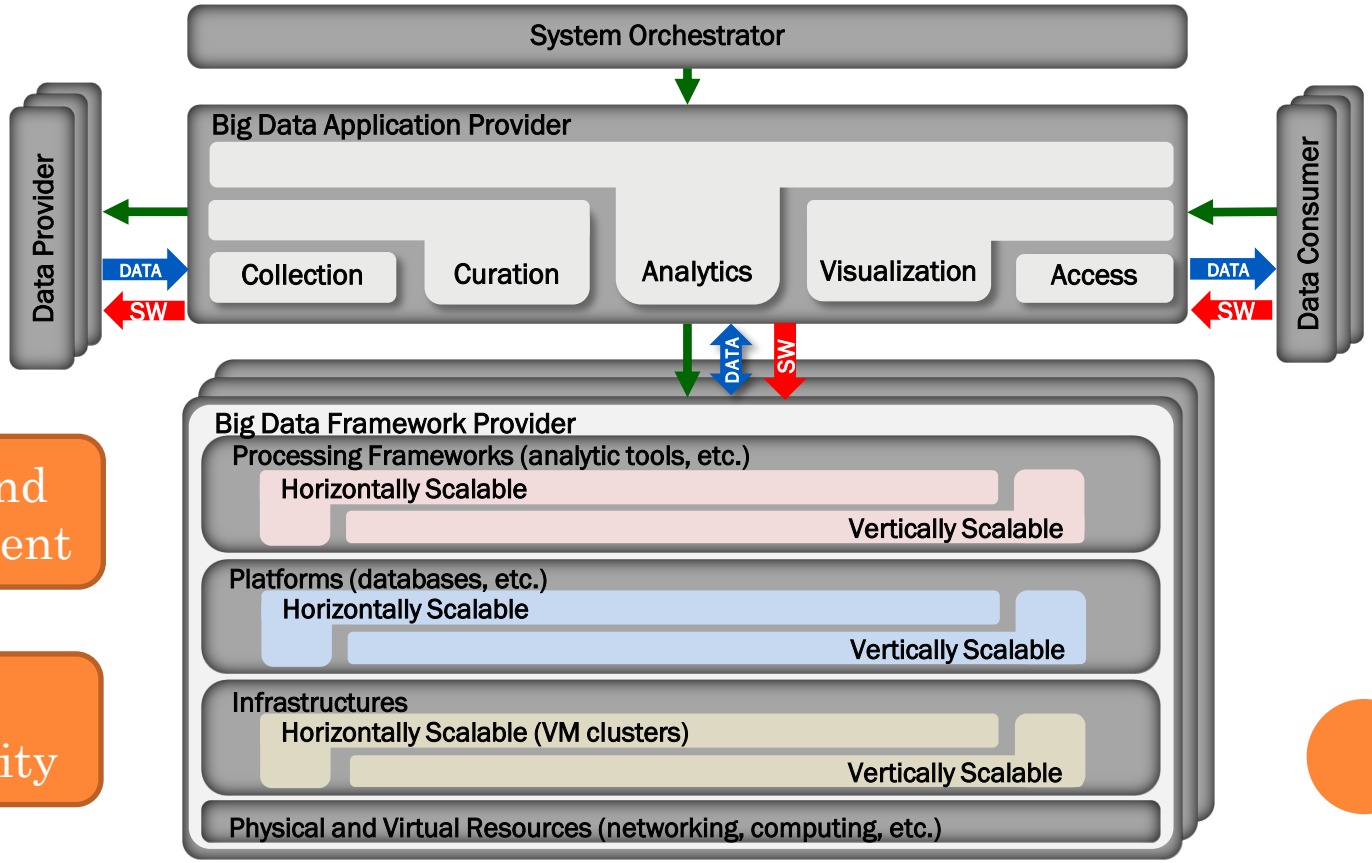




Data Lifecycle Management

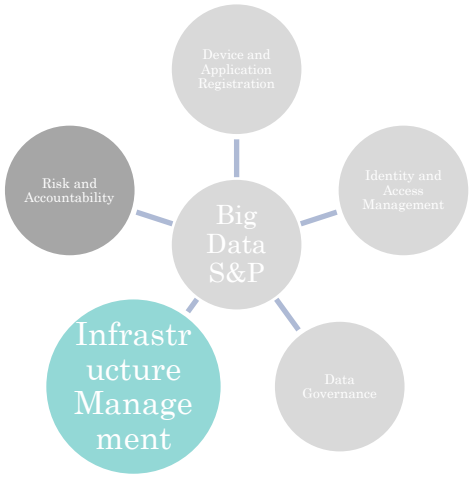
Digital Rights Management

Data Loss Prevention and Detection



Encryption and Key Management

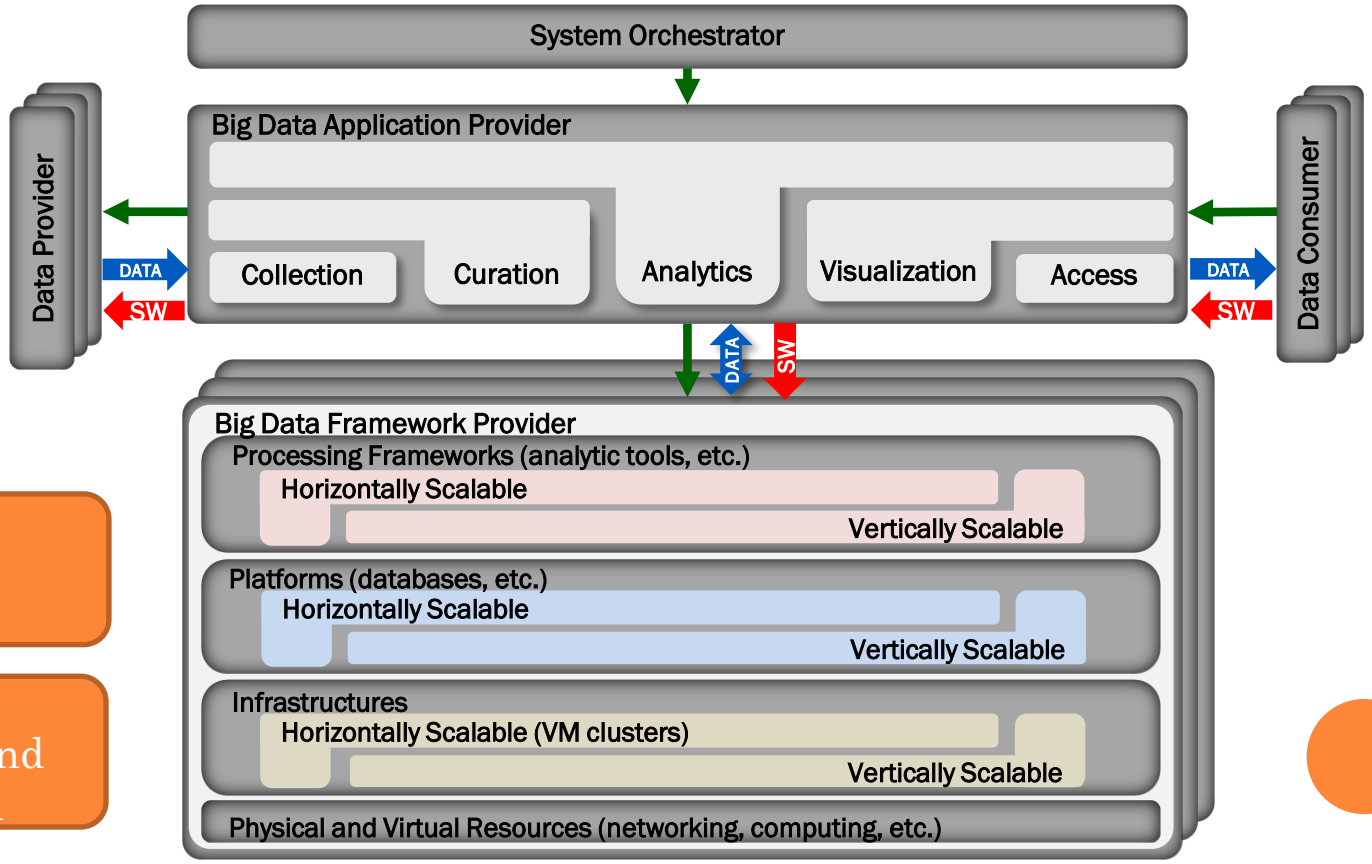
Isolation, Storage Security



Configuration Management

Threat and Vulnerability Management

Network Boundary Control



Logging

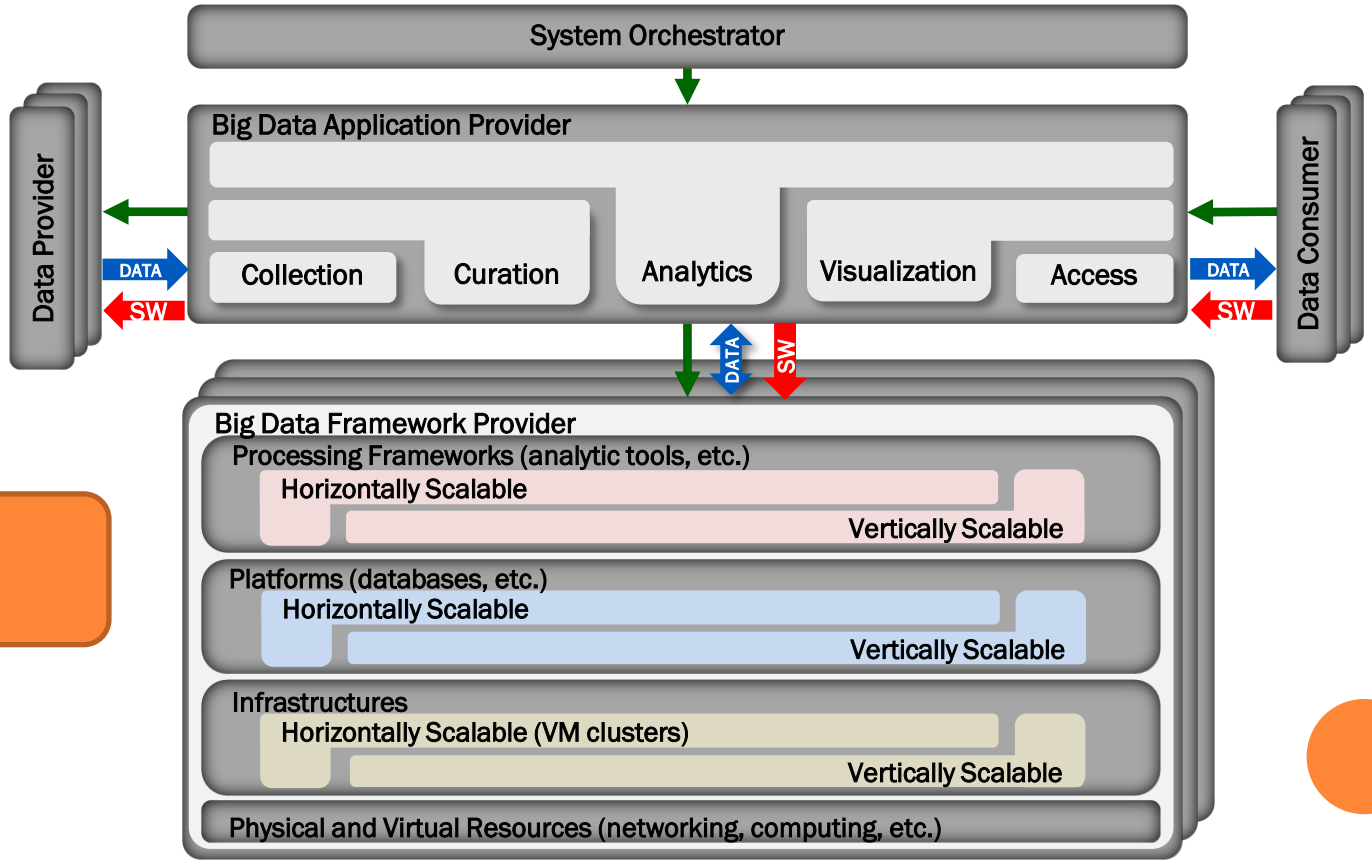
Malware Surveillance and Remediation

Business Risk Model

Compliance

Accountability

Forensics



Emerging Cryptographic Technologies

Utility of Encrypted Client Data	Technology
No operation possible at Cloud	Standard Encryption
Controlled results visible at Cloud	Searchable Encryption - Symmetric - Asymmetric
Transformations possible, but results not visible to Cloud	Homomorphic Encryption
Policy-based Access Control	Identity-based Encryption Attribute-based Encryption

Data Governance: Policy-based Encryption

- Traditionally access control has been enforced by systems – Operating Systems, Virtual Machines
  - Restrict access to data, based on access policy
  - Data is still in plaintext
  - Systems can be hacked!
  - Security of the same data in transit is ad-hoc

- What if we protect the data itself in a cryptographic shell depending on the access policy?
  - Decryption only possible by entities allowed by the policy
  - Keys can be hacked! – but much smaller attack surface
  - Encrypted data can be moved around, as well as kept at rest – uniform handling

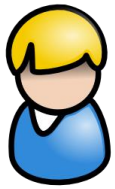
Identity-based Encryption

- Public-Key Encryption

Certificate Authority

Signed Certificate of Public Key

Signed Certificate of Public Key



Alice

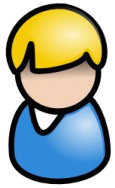
Encrypted Data

Bob

- Id-based Encryption

Master Authority

Master Public Key



Alice

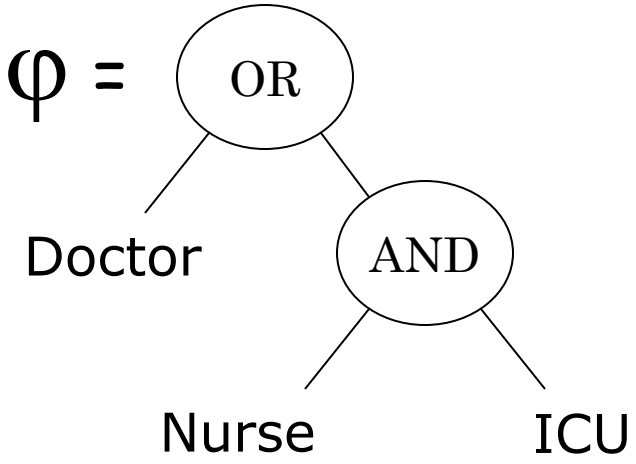
Bob

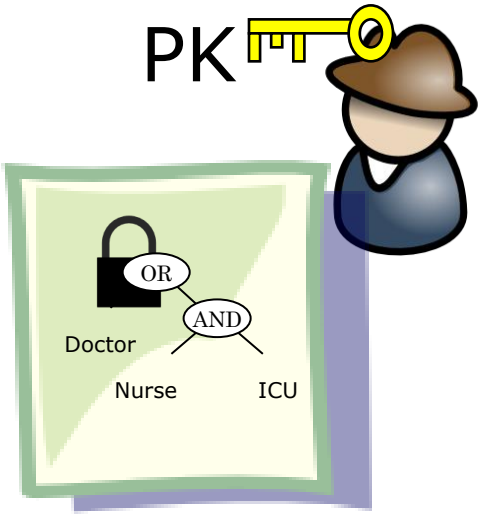
George

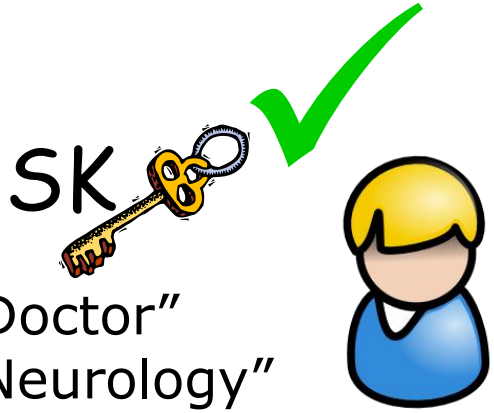
Encrypted Data

Mitchell et al.

Policy-Based Encryption



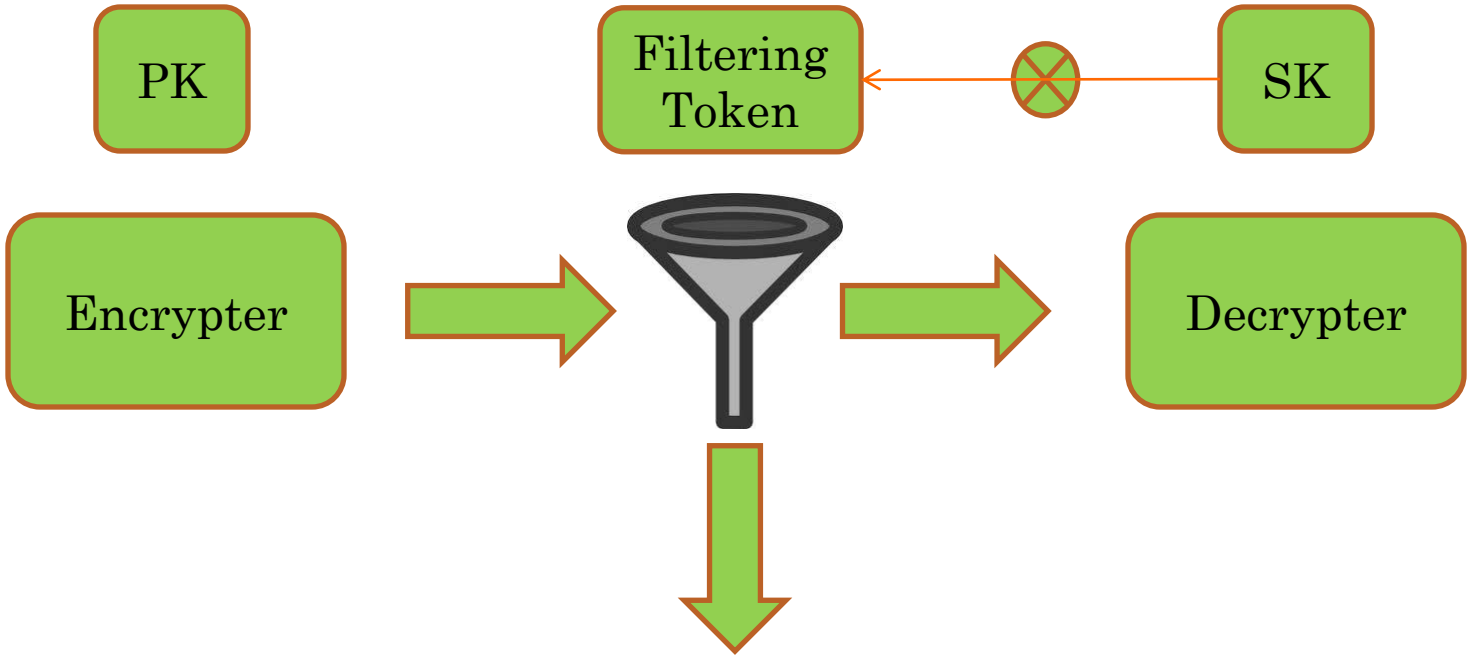




Privacy Preserving Dissemination: Searching and Filtering Encrypted Data

- Suppose you have a system to receive emails encrypted under your public key
- However, you do not want to receive spam mails
- With plain public key encryption, there is no way to distinguish a legitimate email ciphertext from a spam ciphertext!
- However, with recent techniques you can do the following:
  - Give a 'token' to the spam filter
  - Spam filter can apply token to the ciphertext, only deducing whether it is spam or not
  - Filter doesn't get any clue about any other property of the mail!

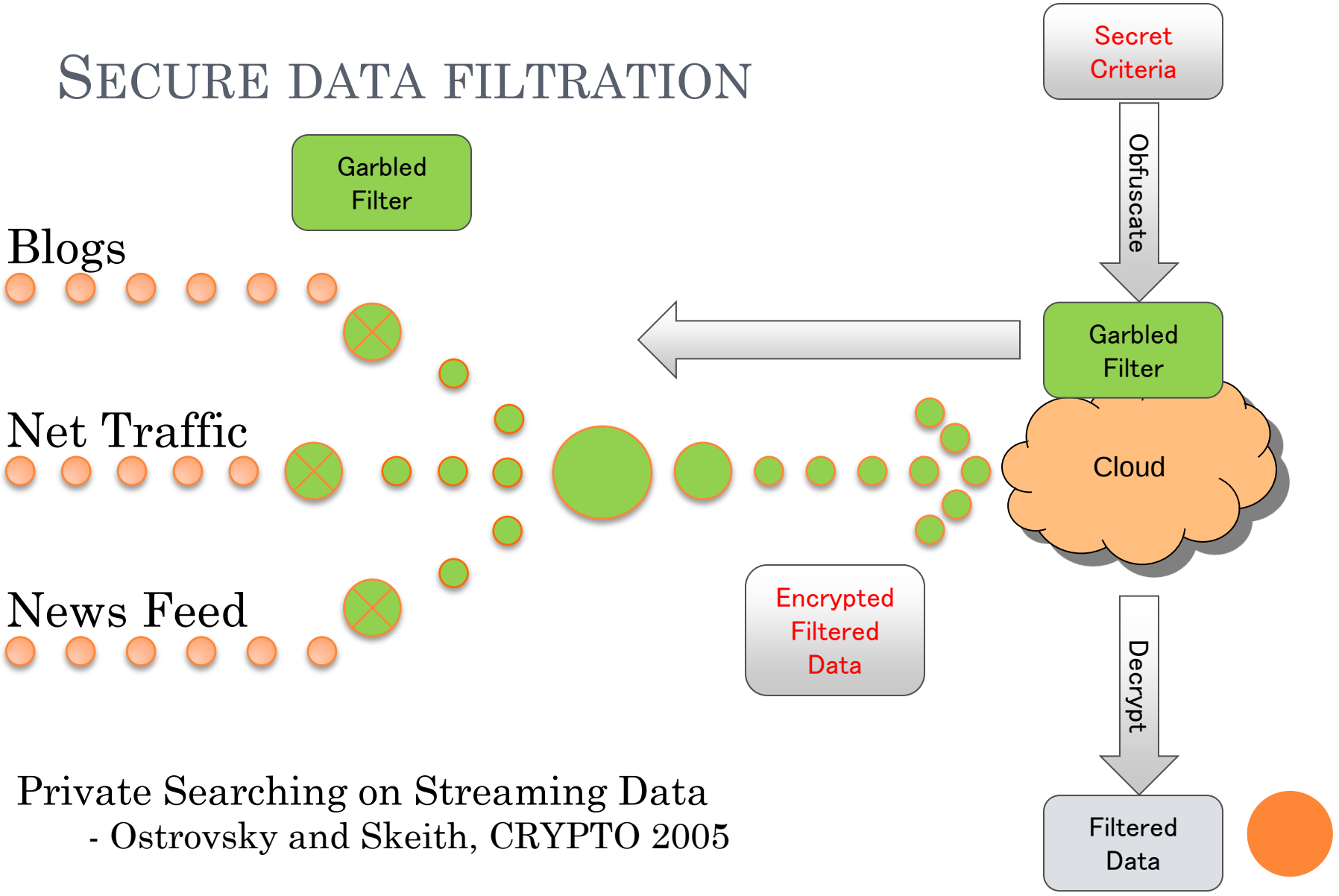
Searching and Filtering Encrypted Data



Secure data filtration

- Problem Scenario:
  - The intelligence gathering community needs to collect a useful subset of huge streaming sources of data
  - The criteria for being useful may be classified – ***private criteria***
  - Most of the streaming data is useless and storing it all may be impractical – ***filter at source***
  - How de we keep the filtering criteria secret even if it is executing at the source?
  - Solution: ***Obfuscate*** the filtration code
    - Even if the source falls into enemy hands, it cannot figure out the criteria

Secure data filtration



Private Searching on Streaming Data
- Ostrovsky and Skeith, CRYPTO 2005

Single Client End-to-End Privacy: Secure Outsourcing of Computation

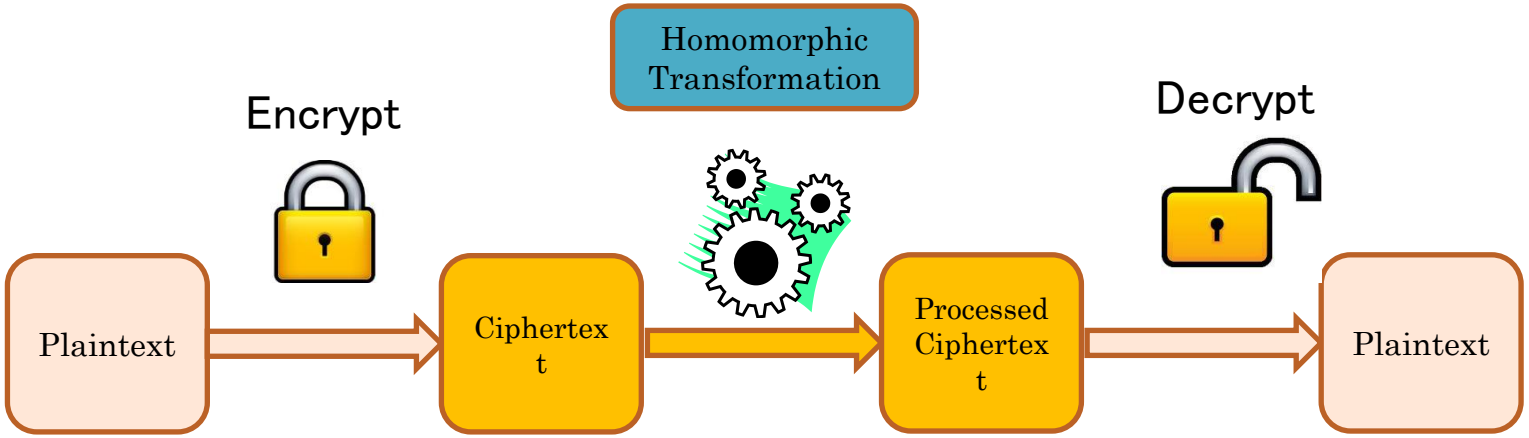
- Suppose you want to send all your sensitive data to the cloud: photos, medical records, financial records, …

- You could send everything encrypted
  - But wouldn't be much use if you wanted the cloud to perform some computations on them
  - What if you wanted to see how much you spent on movies last month?

- Solution: Fully Homomorphic Encryption
  - Cloud can perform any computation on the underlying plaintext, all the while the results are encrypted!
  - Cloud has no clue about the plaintext or the results

Secure Outsourcing of Computation

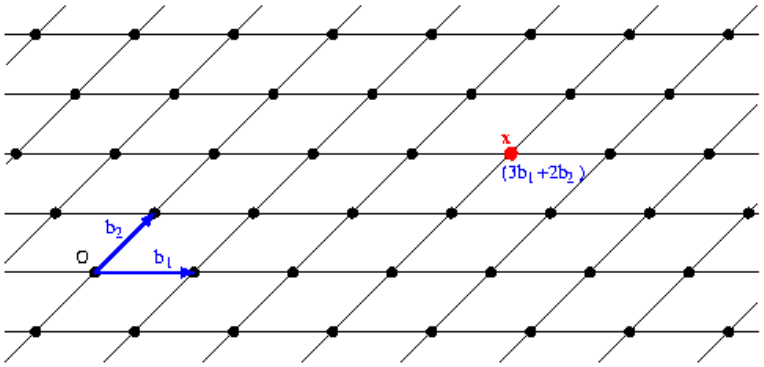
Fully Homomorphic Encryption (FHE)



- With FHE, computation on plaintext can be transformed into computation on ciphertext
- As a use case, a cloud can keep and process customer's data without ever knowing the contents
  - Only customer can decrypt the processed data
  - End to end security of customer data

How does FHE work?

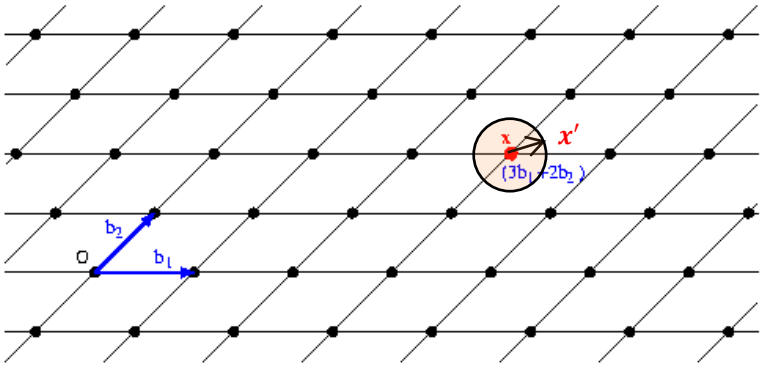
- Intuition:
  - Represent programs as circuits
    - Sequence of additions and multiplications
  - Transform the input data to a high dimensional ring (popularly, lattices)
    - Exploit ring homomorphism with respect to +,×



Source: http://cseweb.ucsd.edu/~daniele/lattice/lattice.html

How does FHE work?

- How do we ensure that the transformed representation hides the plaintext?
  - Solution: add some noise to the representation
  - In sufficiently high dimensions, it is considered hard to derive the closest lattice point, when noise is added
- Now, we have a different problem
  - With each +,×, noise grows!
  - At some point, data may be irrecoverable
  - Solution: noise reduction techniques
    - Bootstrapping, Modulus switching



Source: http://cseweb.ucsd.edu/~daniele/lattice/lattice.html

Thanks!